CONTROL DATA

CORPORATION

BULLETIN

APPLICATIONS DEVELOPMENT

6400/6500/6600
ASCENT-TO-COMPASS TRANSLATOR

6400/6500/6600
ASCENT-TO-COMPASS TRANSLATOR

CONTENTS

I ASCENT-TO-COMPASS TRANSLATOR

The ASCENT-to-COMPASS Translator provides compatibility between COMPASS and ASCENT assembly systems. The translator will run under the 6400/6600 SCOPE Operating System, Version 2.0 or Version 3.0. The translator requires a job field length of 35000.

The translator processes input until an end-of-file is found. Input may be in BCD source format or in ASCENT COSY format. Complete translator listings may be obtained; or, if requested, only headings which indicate subprograms processed, and lines with diagnostics will be listable output. Translator output is in COMPASS input form (BCD source card images). On option, translator output may be formatted for subsequent input to EDITSYM before assembly by COMPASS.

II INPUT/OUTPUT OPTIONS

The control card for the translator is as follows:

TRANS (list of input/output options)

The total number of options listed must not exceed three, one each for input, listable output, and compile output. File names may not be I, Y, L, E, or C, since these are used for option specification.

Input Options

I=fname   Input consists of BCD source card images on file fname. fname is assumed INPUT if I=0 or I alone appears in the option list.

Y=fname   Input is in ASCENT COSY format on file fname. fname is assumed INPUT if Y appears alone or if Y=0.
Y must appear if COSY input is used.
The COSY file will be rewound by the translator before reading begins.

If neither I nor Y is used, I-type input is assumed on file INPUT.

Output List Option

L=fname   Listed output will be on file fname. If L appears alone or is absent, listings will be on file OUTPUT.
If L=0 translator listing is suppressed except for headings to indicate subprograms processed, and lines with diagnostics.

Compile Output Options

C=fname   Translator output will be COMPASS input (BCD source programs) with no EDITSYM control cards added. C alone or C=0 causes output to be written on a file named COMP.

E=fname   Translator output will be BCD source programs formatted for subsequent input to EDITSYM before assembly by COMPASS. If E appears alone fname is COMP.
If E=0, C-type output will be written on file COMP.

If C and E are both missing, C-type output will be written on file COMP. Either E or C may appear, but not both.

If E has appeared, the translator will insert EDITSYM controls between each subprogram translated:

> When ASCENT or ASPER card is encountered, the following card image
> will be inserted before the IDENT card produced by the translator:
>
> *DECK,dname
>
> dname is taken from the ASCENT or ASPER card address field. If no name appeared
> on the ASCENT or ASPER card, the line will read *DECK.

Upon encountering an ASCENT/ASPER END card, the translator will output the END card followed immediately by a *END card.

E output should not be requested for mixed FORTRAN/ASCENT decks, since the translator does not recognize the beginning and end of input that is not headed by an ASCENT or ASPER card. The translator will insert the appropriate EDITSYM control cards before and after ASCENT subprograms, but not FORTRAN programs.

The translator will produce output for all input received. ASCENT/ASPER programs are translated into COMPASS; anything not recognized is simply reproduced. Each ASCENT or ASPER card begins the actual translation process, which continues until an END card is found. An end-of-file terminates all translator processing.

For an ASCENT subprogram to be recognized, the ASCENT or ASPER instruction must begin in column 11 or beyond, preceded by blanks, and must be followed by at least one blank column.

The translator will produce COMPASS subprograms, which, when assembled, should produce results identical to those of ASCENT. No reliability is guaranteed for ASCENT code with errors.

The normal listing produced by the translator contains a listing of each entire ASCENT subprogram as it was read followed by a listing of the new subprogram as it has been translated. Code which was not translated but merely passed on to the output file will be indicated by a page heading, and only the first card will be listed.

A sequence number for each line appears to the left side of the ASCENT card images. This number will match, line-by-line, a sequence number listed for the COMPASS code which follows. The number is not produced on the translator output compile file; it is listed merely for convenient cross-referencing of the translator output.

When COSY input is indicated, the file will first be rewound. When the translator is finished, the new compile file will be rewound. Non-COSY input and listable output are never rewound.

III TRANSLATOR CAPABILITIES

The ASCENT-to-COMPASS Translator will perform the following functions:

1. Replace operation codes:

ASCENT	COMPASS
ASCENT	IDENT
ASPER	IDENT
	PERIPH
DPC	DATA
BCD	DATA
BCI	DATA
CON	DATA or VFD
IFF	(Multiple)
IFZ	IFEQ
IFN	IFNE
COMMON	(Multiple)
ORGR	ORG

2. Change the address fields of some pseudo operations:

$$\text{IFZ } v_1, v_2 \quad \text{becomes} \quad \text{IFEQ } v_1, 0, v_2$$
$$\text{IFN } v_1, v_2 \quad \text{becomes} \quad \text{IFNE } v_1, 0, v_2$$

VFD: remove A, D, and N notations; add COMPASS notations for display code subfields.

CON: create VFD if any item is other than absolute numeric.

DPC: nncc...cc becomes DATA nnHcc...cc
        cc...cc becomes DATA H*cc...cc*

BCD, BCI: characters are converted to octal equivalents.

COMMON: convert to COMPASS instructions sequence:

        BSS   0

        USE   /name/

p_1 BSS n_1

p_2 BSS n_2 p_n are array names

        . n_n are lengths

        .

        .

p_n BSS n_n

        USE   *

LIST: if v_1 is not numeric, the translator can not know its value; there-
        fore it will produce the following sequence of instructions for
        LIST v_1:

            IFEQ v_1,,1

            LIST L

            IFNE v_1,,1

            LIST -L

        if v_1 is numeric and has a zero value only the following card will
        be produced:
            LIST L

        if the value is non-zero, the translator will produce:
            LIST -L

3. Analyze the format of address expressions; convert them if necessary so they will
    produce the same result in a COMPASS assembly.

4. Replace ASCENT field and subfield separators with COMPASS separators: only blanks
    will be used between fields, only commas will be used between subfields where either
    comma, blank, or = is legal in ASCENT. In CP 3-letter register instructions, a comma,
    blank, or = will be replaced by +, except if the instruction is LXi, AXi, NXi, ZXi, PXi,
    or UXi.

5. Replace C in column 1 with *. Insert * in column 1 of any card on which a period is the
    first non-blank character.

6. Re-format other instruction lines as necessary; left justifying the location field entry in
    column 2; move the comments field to column 36 when operation code and/or variable
    field are empty; look for and remove characters . and $ when they are not at least one
    character removed from location, operation, or variable field.

7. Re-format MACRO lines from:
            MACRO name, $p_1, p_2, \ldots, p_n$
    to: name   MACRO $p_1, p_2, \ldots, p_n$

8.   Find formal macro parameters in the operation code fields within a macro definition and re-structure that operation code.

9.   Replace ASCENT literals with notations which produce equivalent results in COMPASS.

10.   If necessary, re-order the subfields of CP jump instructions so that registers are first, operands are last.

11.   For PP instructions SHN, xJN, if the operand value is numeric and greater than 37B, analyze and revise the operand value so that it appears as -n.

IV SYMBOL CREATION

In some cases the translator will create symbols for consistent results. They will be of the form C.nnnnn, where nnnnn is unique for each symbol created in the subprogram. Symbols are created for the following reasons:

1. An ASCENT address expression must be re-formatted to produce equivalent results in COMPASS. For example, the expression A+B*C is evaluated in ASCENT from left to right with no significance attached to the operators. COMPASS would evaluate A+B*C as if it were A+(B*C). To compensate, the translator creates a line:

        C.nnnnn   EQU   A+B

    and inserts

        C.nnnnn*C

    in place of the original expression.

    When the expression occurs in an EQU, the created symbol line is inserted immediately before the EQU since EQU requires that symbols in its address field be previously defined. When the expression occurs in any other instruction, the created line is saved by the translator for the end of the program to assure that all symbols involved in the expression have been defined.

    Examples of lines in an ASCENT program:

             SA1   A+B*C-D/E
        XXX   EQU   Y+Z-X*2

    The translator will produce the following lines:

        C.00000   EQU   A+B
        C.00001   EQU   C.00000*C-D
        C.00002   EQU   Y+Z-X

    Lines defining C.00000 and C.00001 would appear immediately before the END card.
    The original instruction becomes:

             SA1   C.00001/E

    The line defining C.00002 would appear immediately before the original EQU instruction line, such as:

        C.00002   EQU   Y+Z-X
        XX   EQU   C.00002*2

2. Literals might cause a created symbol to be defined. A literal specifying a single numeric constant (octal, decimal, single or double precision) will be replaced with the COMPASS equivalent =n. A literal specifying a single symbol will be replaced by =Sname. Some other ASCENT literals will cause a created line to be produced. It will appear at the very end of the program.

Example:

                        SA1   (1.0,2.0)   ASCENT complex constant literal

would be converted to

                        SA1   C.00000

just before the END card would appear

        C.00000   VFD   60/1.0
        C.00001   VFD   60/2.0

3. In some cases, the translator must produce multiple card images to reproduce the intent of an ASCENT instruction, as in the case of IFF. Created symbols will be used as bracket names on a COMPASS conditional instruction and its ENDIF. For example, consider the translation of the following lines:

                        IFF   ALPHA,BETA,GAMMA
                        VFD   N30/0,D30/THETA

The translator is not assembling the code and does not know the value of ALPHA. It must produce the following set of COMPASS instructions:

C.00000	IFNE	ALPHA	(Skip to C.00000 ENDIF if ALPHA is zero)
	IFEQ	BETA, GAMMA,1	(Assemble next line if ALPHA was non-zero and BETA=GAMMA)
	VFD	30/0,30/0HTHETA	
C.00001	IFNE		(skip to end of code sequence)
C.00000	ENDIF		
	IFNE	BETA,GAMMA,1	(Assemble next line if ALPHA was zero and BETA≠GAMMA)
	VFD	30/0,30/0HTHETA	
C.00001	ENDIF		

V PROGRAMMER RESPONSIBILITY

Certain translations must be made by the programmer. There will be no ASCENT-to-COMPASS conversion in the following cases.

1. Macro formal parameters in <u>address fields</u> within a macro definition prototype will not be given special attention. Thus, when parameters are combined with register letters, as in XPPP, they will not be properly processed by COMPASS. They must be changed manually to conform with COMPASS rules. For example, XPPP+X1 could be changed to X.PPP+X1 or to X → PPP+X1.

    However, formal parameters in the <u>operation code</u> field within a prototype will be properly translated if they conform to ASCENT rules:

    The formal parameter is the entire operation code.

    The formal parameter name begins with the second letter of the operation code.

    The parameter name begins with the third letter of the operation code.

    The translator will flag each macro definition on the output listing so it will be easily distinguished.

2. The program name should be on the ASCENT or ASPER card, not on the first ENTRY card as ASCENT allows. A diagnostic will be produced if there is no name on the ASCENT or ASPER card.

3. In some cases during a central processor assembly, ASCENT interprets the address element * to mean the last P-counter value instead of the current P-counter (location counter) value. Without assembling the ASCENT code, the translator cannot always compensate for this, and will not attempt to do so. In COMPASS, the value of * will always be the current location counter value. A difference of 1 may result when * is used in an instruction expression after a full word of data has been assembled but before the next word has any data generated for it. In this case, ASCENT has not yet advanced the P-counter; COMPASS has.

    Example:

TABLE	CON	ALPHA
	CON	BETA
	CON	GAMMA
TABLEN	EQU	*-TABLE

    The value of TABLEN in an ASCENT CP assembly will be 2. (It would be 3 in a PP assembly.)

The translated code will appear as:

```
TABLE        VFD    60/ALPHA
             VFD    60/BETA
             VFD    60/GAMMA
TABLEN       EQU    *-TABLE
```

The value of TABLEN in a COMPASS assembly will always be 3, since the location counter has already been advanced when the EQU instruction is processed.

In the following examples, though, ASCENT and COMPASS will produce the same value for *:

```
a.   +          SA1    TABLE
     HERE       EQU    *

b.   +          SA1    *
```

The translator will issue a warning flag when an * is used as an operand in the address field of an EQU, BSSZ, or BSS instruction. It is the responsibility of the programmer to adjust the address expression to produce the same value in COMPASS as in ASCENT, if necessary.

4. An IFF statement immediately following an IFF will not be translated. The translator must produce multiple card images for each IFF and cannot do this for nested IFF's. A diagnostic will be produced if nested IFF's appear.

5. Comments cards which might appear before the ASCENT or ASPER instruction will not be translated to COMPASS comments cards.

VI DIAGNOSTICS AND WARNINGS

Diagnostics

At the end of the listing of each COMPASS subprogram produced by the translator, there will be a list of diagnostics and/or warnings for the programmer.

Diagnostics will be produced when any of the following is encountered:

No name on ASCENT or ASPER card.

End-of-file is found before an END card while translation is in process. The translator will supply an END card with a blank location field and blank address field.

Unrecognized operation code diagnostic will appear for any but the following legal operation codes:

    CP or PP machine mnemonic

    ASCENT 2.0 pseudo instruction

    Macro call

    Formal parameter name inside a macro definition

    Octal operation code of the proper format

    Operation code inside a macro definition which is partially a formal parameter name

IFF following an IFF. An * will be inserted in column 1 of the second IFF to make it a COMPASS comment card, and a diagnostic message will appear.

Translator's Created Symbol Table is full. This table holds 50 created symbols of the type which have to be saved for output at the end of assembly. Some of the statements requiring symbol creation should be removed from the ASCENT program.

If any of the above conditions occur, the line containing them will be flagged with a number. This number refers to a printed description of the error at the end of the program.

Warnings

In addition to the diagnostics listed above, warning messages will be produced at the end of the COMPASS listing for any of the following conditions. The line on which each condition occurs will be flagged with a row of asterisks. Most warnings serve as reminders that a change in coding practice will be required of assembler users.

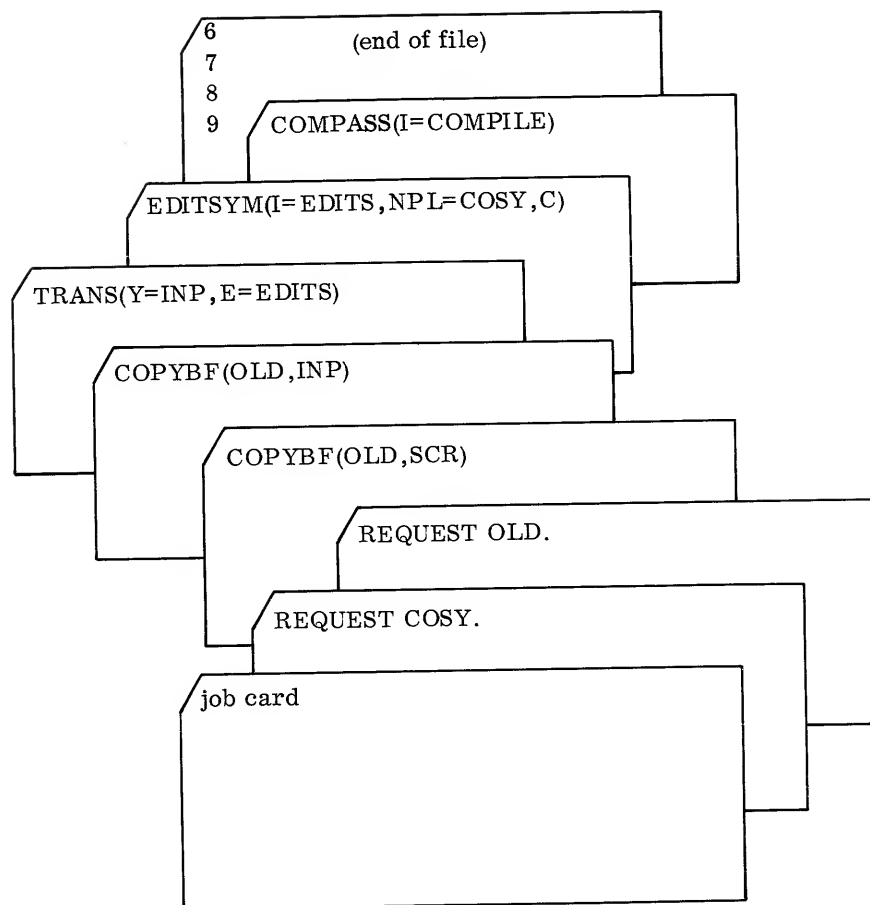|	
In an ASPER program, a shift instruction (SHN) or no-address jump (xJN) with an address value greater than 37B	COMPASS requires these negative jumps (or shifts) to be expressed as -n. The translator will convert all such numbers greater than 37B, but the warning flag will be given.
BCD or BCI pseudo instruction	COMPASS has no BCD facility. BCD character strings in an ASCENT program being translated will be converted to octal equivalents, and a warning will be given.
CON with an address expression	In COMPASS, the nearest equivalent to CON is DATA, but DATA allows only absolute numeric or character values. A 60-bit address value must be expressed with VFD. Translator will convert each expression to: VFD 60/expression
DPC pseudo instructions	Since DPC in ASCENT is not exactly equivalent to the COMPASS DIS instruction, this warning will be issued. However, DPC will be translated to produce proper results in a COMPASS assembly.

If macro definitions do occur, the programmer will be reminded that he should check for legal COMPASS use of formal parameter names.

If the character * is used as an element in the address expression of an EQU, BSSZ, or BSS, a warning will be given.

VII EXAMPLES OF JOB SET-UPS
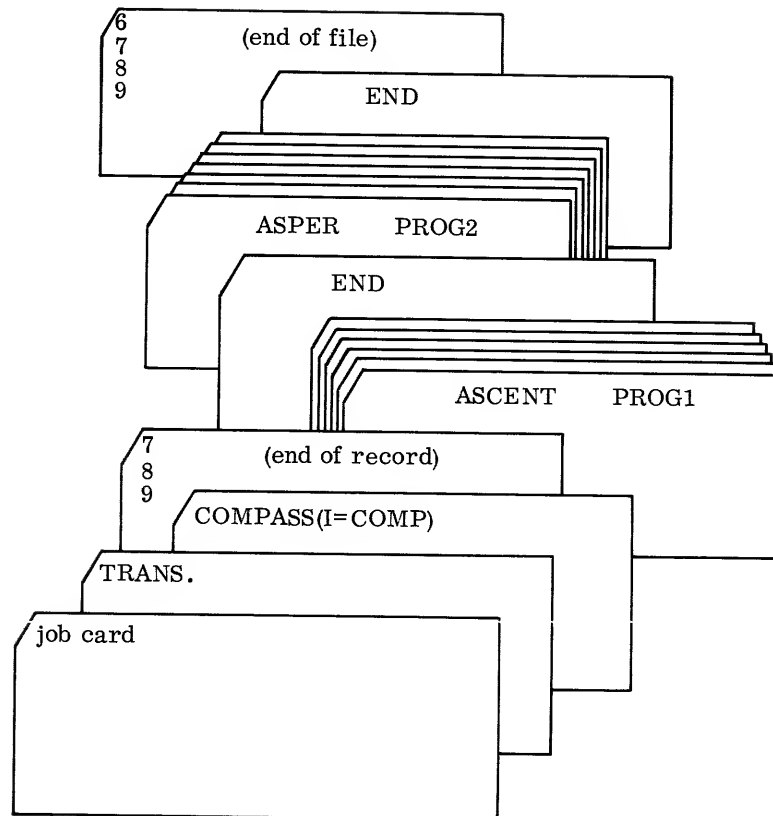
The above deck shows the control cards to translate BCD source subprograms on the INPUT file, produce translator listings on file OUTPUT, produce translator compile output on the file COMP, and assemble the translated COMPASS program.

```
 6                    (end of file)
 7
 8
 9      COMPASS(I=COMPILE)

     EDITSYM(I=EDITS,NPL=COSY,C)

   TRANS(Y=INP,E=EDITS)

     COPYBF(OLD,INP)

       COPYBF(OLD,SCR)

         REQUEST OLD.

       REQUEST COSY.

   job card
```

The above deck shows the control cards to: translate the COSY decks which are the second file of tape OLD, produce EDITSYM input on the file EDITS, process the file through EDITSYM and produce compressed symbolic on file COSY, produce COMPASS compiled input on the file COMPILE, and assemble the translated COMPASS program.